



# Web Application Toolkit for Squeak

# Introduction

## ➤ About us

- Lukas Renggli
  - Adrian Lienhard
- } users of the Seaside framework

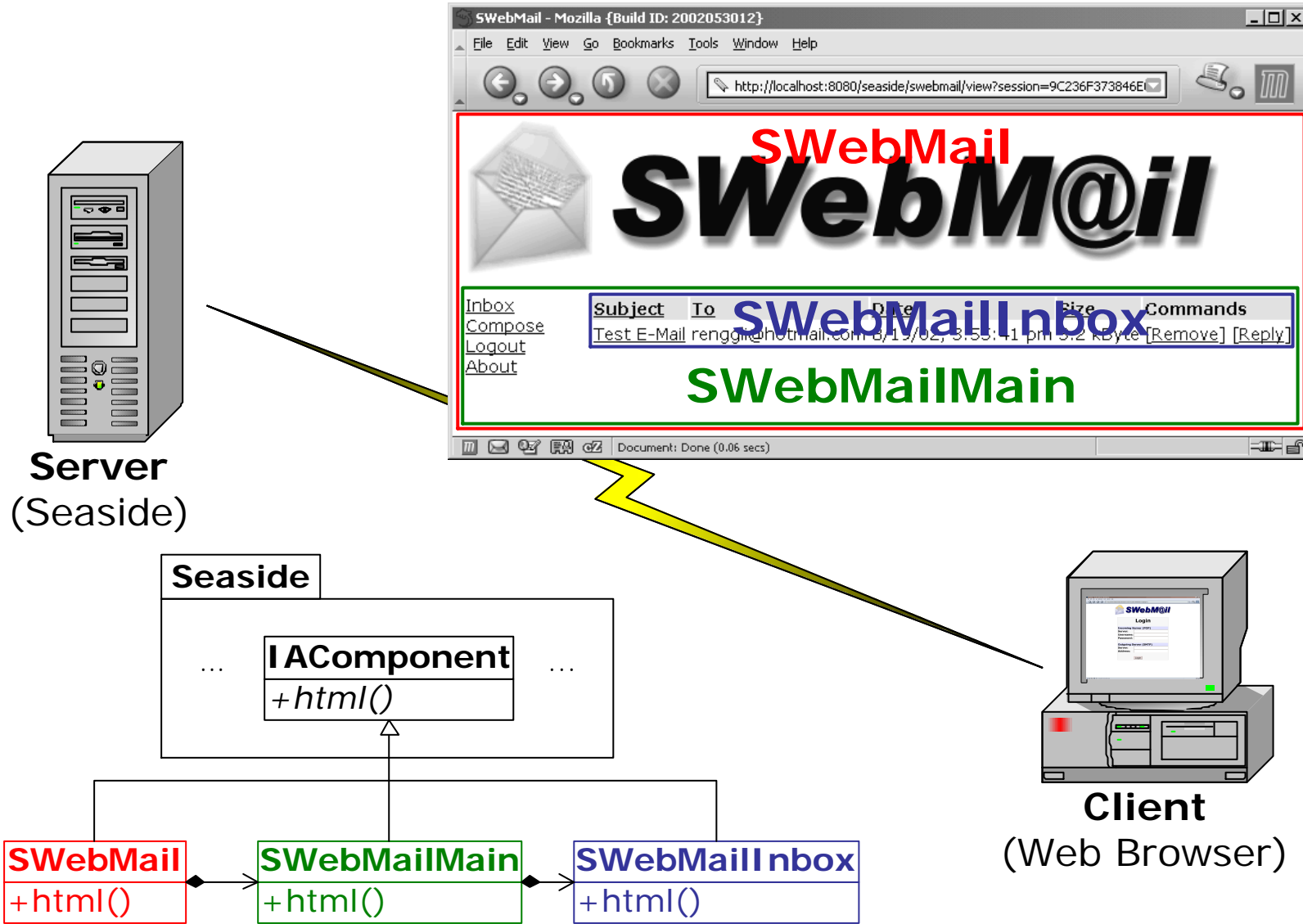
## ➤ Seaside Developers

- Avi Bryant
- Julian Fitzell

# Content

- What can be done with Seaside?
  - Example Application: SWebMail
    - Reading mailbox from POP account
    - Sending e-mails over SMTP server
- Why should Seaside be used?
  - Complete separation of logic and design
  - Object-oriented programming on the fly
  - Application control flow
- Why is Seaside so easy to use?
  - Action methods
  - Bidirectional binding of properties
  - Session management
  - ...

# Component-Template Model

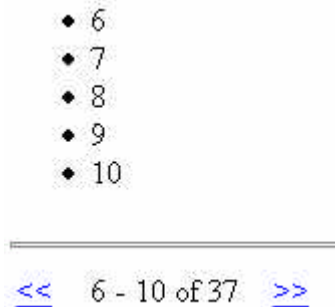


# Reuse by Composition

➤ Component architecture

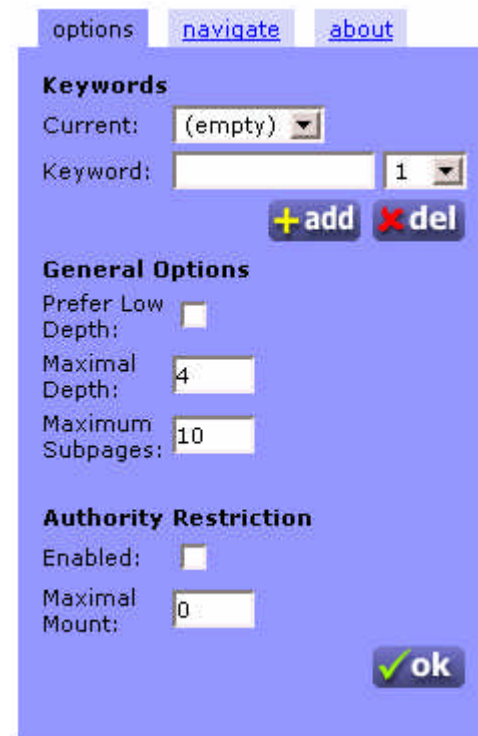
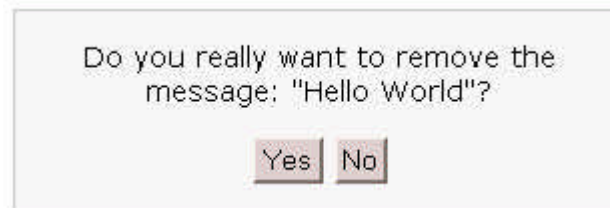
➤ Widgets

- Tab sheet
- Date selector
- Batches
- Sortable tables



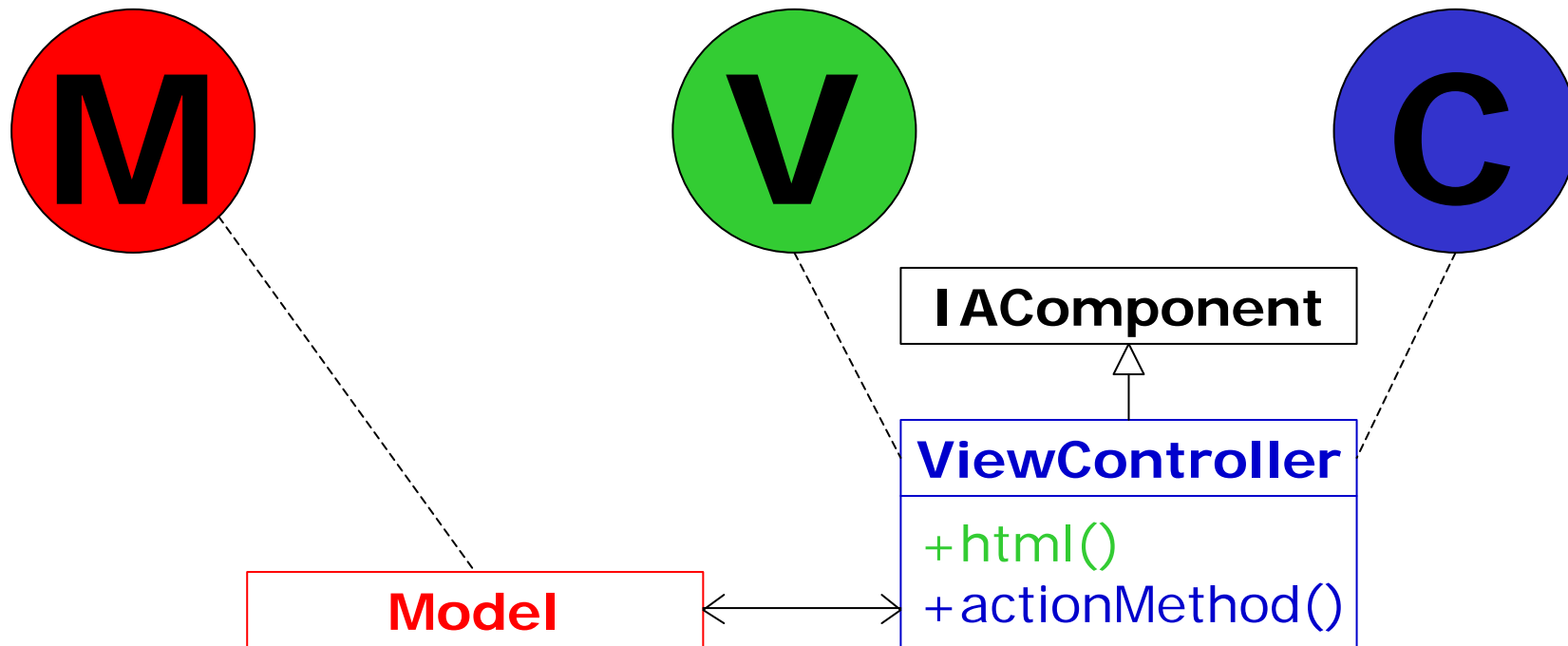
➤ Dialogs

- Information
- Confirmation
- Choose
- Ask



# Separation of Logic and Design

- Programmers should create *logic*
- Designers should create *HTML (templates)*



# Creation of Templates

- Override `IAComponent>>html` and return
  - a string containing valid HTML

```
SWebMail>>html
^ '<html>
  <head>
    <title>SWebMail</title>
    <style type="text/css">[css]</style>
  </head>
  <body>
    <h1>SWebMail</h1>
    <swemailmain sea:id="main"></swemailmain>
  </body>
</html>'
```



- External files
  - WYSIWYG editors
  - Designers can work independently

# Binding of Properties

## ➤ Unidirectional binding

```
SWebMailView>>html
^ '...
  <tr>
    <td>Subject:</td>
    <td>[message.subject]</td>
  </tr>
  ...'
```



## ➤ Bidirectional binding

```
SWebMailCompose>>html
^ '...
  <form sea:id="reload">
    <input type="text" sea:id="subject">
  ...'
```



```
SWebMailCompose>>subject
^ subject ifNil: [ subject := String new ]

SWebMailCompose>>subject: aString
subject := aString
```



# Template-Variable Look Up

- Looking up sequence of properties
  - Template Locals
  - Accessor methods
  - Instance variables

```
SWebMailInbox>>html
^ '...'
  <tr sea:id="item/sortedmessages">
    <td>[item.subject]</td>
    <td>[item.fromAddress]</td>
    ...'
```



# Action Methods

- Modify the component's state
  - Message is executed
  - Same component is redisplayed
- The action method might also
  - Expect a parameter
  - Replace the current component at runtime
  - Throw an exception: a stack-dump will be sent to the browser

```
SWebMailMain>>html
^ '...'
  <a href="@about">About</a>
  ...'
```

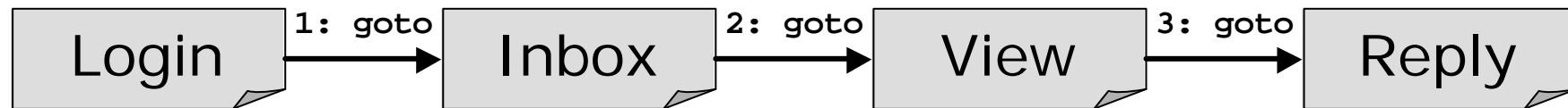


```
SWebMailMain>>about
self callPage: (SWebMailAbout new)
```



# Application Control Flow

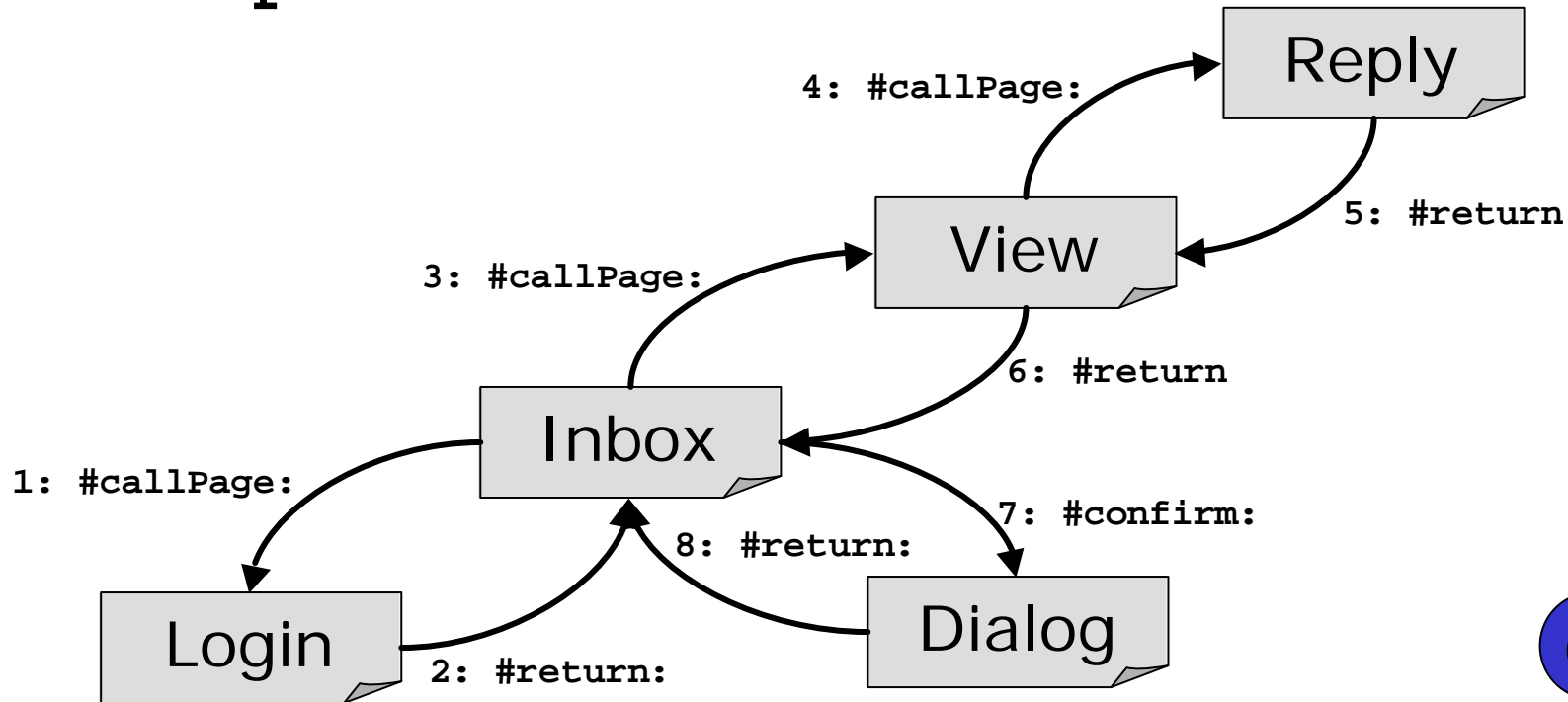
- Common way
  - Manual stack handling
  - Difficult to implement
  - Error prone



- Seaside
  - Transforms traditional Smalltalk send/return control flow onto the web
  - Supports
    - Backtracking (back button)
    - Transactions (full blocks)

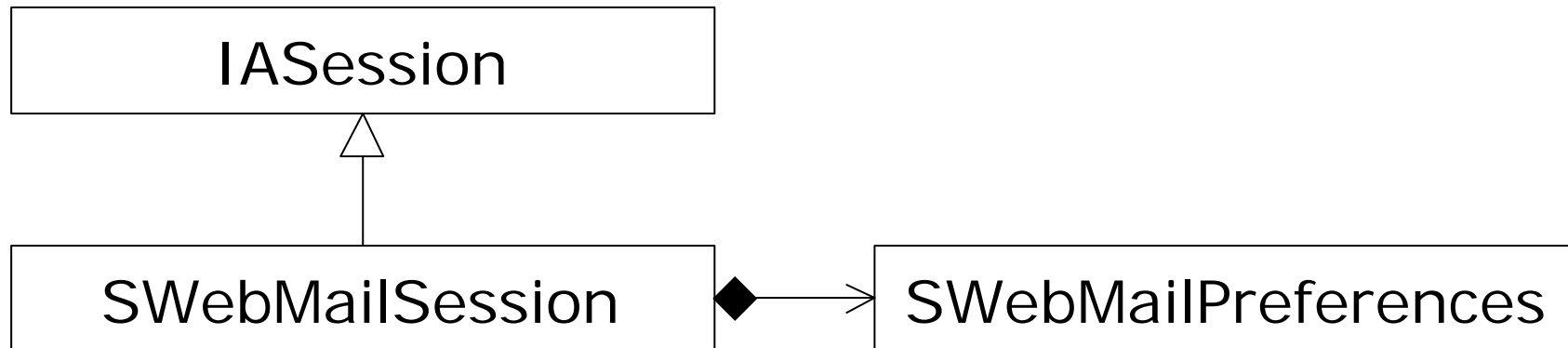
# Application Control Flow

- Call pages with message:  
**IComponent>>callPage: aComponent**
- Return result from page with message:  
**IComponent>>return: aValue**



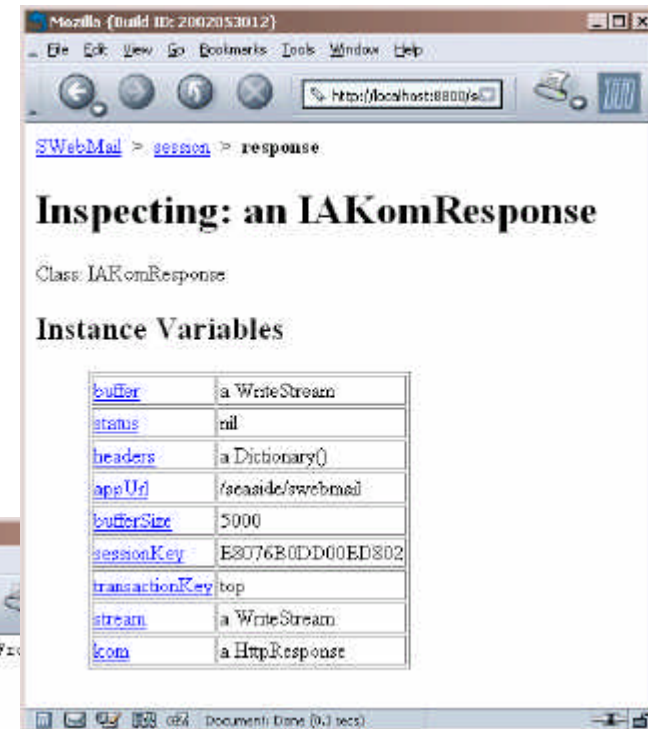
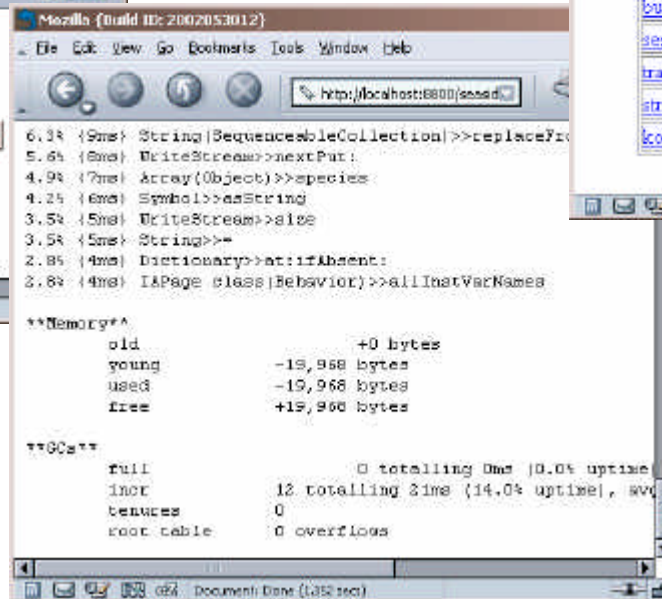
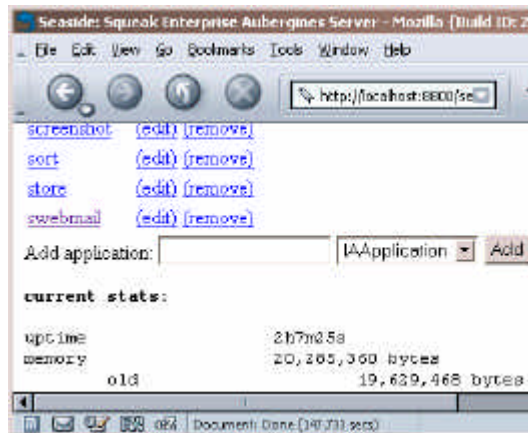
# Session Management

- Seaside manages and caches all needed session data automatically
  - Each session has its own set of objects
  - Caches: LRU, Expiring, ...
- Example: IASession is subclassed to
  - Store the user-preferences
  - Login and logout handling



# Goodies

- Web-based inspector
- Profiler
- Configuration tools



# Seaside 2.0 (Borges)

- Full reimplementaion
- Supports a wider set of styles
- Layered design



## **Borges-Components**

Control flow between pages  
Safe backtracking  
Component embedding

## **Borges-Support**

Callback system  
HTML renderer

## **Borges-Kernel**

HTTP session  
Request  
Response

# Comparison

	<b>Seaside</b>	<b>Zope</b>
<b>Language</b>	Smalltalk	Python
<b>OOP</b>	Complete	Partly
<b>Reuse</b>	Heavily	Possible
<b>Separation</b>	Yes	No, but ZPT
<b>Control Flow</b>	Message calling	Request, Response
<b>Debugging</b>	WebInspector	Difficult
<b>Database</b>	(Seashell), GOODS, Tantalus, ODBC, ...	ZODB, PostgreSQL, MySQL, ODBC, ...
<b>Large Systems</b>	Not tested	Possible
<b>Integration</b>	Apache	Apache

# Summary

- Seaside: Squeak Enterprise Aubergines Server
  - Developers: Avi Bryant, Julian Fitzell
  - Current Version: 0.94.1
  - License: Squeak-L, BSD
  
- Works best with
  - Squeak 3.0, 3.1, 3.2
  - Comanche 4.9, 4.10
  - Windows, Mac OS X, Linux, SUN Unix, ...
  
- Ports underway
  - Dolphin Smalltalk
  - VisualWorks

# References

## ➤ Seaside

- Homepage

<http://www.beta4.com/seaside>

## ➤ Seaside 2.0 (Borges)

- Documentation

<http://beta4.com/squeak/aubergines/docs/seaside/borges.html>

- Download

<http://beta4.com/squeak/aubergines/source/Borges.st>

## ➤ SWebMail

- Homepage

<http://renggli.freezope.org/programming/smalltalk/swebmail>

- Running Server

<http://aurora.unibe.ch:8080/seaside/swebmail>



Lukas Renggli, Adrian Lienhard